Wireless Pers Commun (2018) 100:213–225
https://doi.org/10.1007/s11277-017-5061-y

Energy-Aware and Reliability-Aware Mapping for NoC-Based Architectures

Naresh Kumar Reddy Beechu[1] · Vasantha Moodabettu Harishchandra[1] · Nithin Kumar Yernad Balachandra[1]

Published online: 20 November 2017

Abstract Extensive research has been conducted on task scheduling and mapping on a multi-processor system on chip. The mapping strategy on a network on chip (NoC) has a huge effect on the communication energy and performance. This paper proposes an efficient core mapping for NoC-based architectures. Which focus on energy- aware and reliability-aware mapping issues for NoC-based architectures and considers new applications with insignificant inter-processor communication overhead to be added to the system. This methodology was assessed by applying it to various benchmark applications. Simulation results reveal that the proposed mapping algorithm greatly improves the reliability of the system and reduce the communication energy.

Keywords System on chip · Core · Network on chip · Communication energy

1 Introduction

With the technological advancements in integration of chips, system on chip (SoC) has become a major pattern in popularized items since 1980. Numerous cores may be integrated on a single chip [1]. Numerous researchers have been designing multi-core architectures. Moreover, as the complexity of applications running on a chip is extremely high, it has to support high bandwidth communication and high resolution. Hence communication between processors/cores becomes more critical. Therefore, Networks-on-Chip (NoC) may be an effective interconnect solution for complicated chips comprising of the many heterogeneous intellectual property (IP) cores [2]. The NoC solution involves a networking approach to on-chip communication and considerably enhances reliability and performance compared with the conventional bus- based structures (e.g., AMBA bus). However, it is currently difficult to fulfill both energy and reliability requirement during faults at mapping cores on an NoC.

✉ Naresh Kumar Reddy Beechu
naresh.nitg@gmail.com

[1] National Institute of Technology Goa, Farmagudi, India

In this paper, Energy and Reliability Aware Mapping (ERAM) algorithm is proposed, where cores are mapped on a NoC platform with respect to communication energy. When the fault occurs in processing cores, tasks are efficiently migrated from the failed core to the nearest free core with respect to minimum communication energy. Simulation results reveals that not only reduction in communication energy in the ERAM but it also the effectiveness of ERAM in terms of computation time, reliability, and throughput [3].

The remaining of this paper is organized as follows. A literture review is presented in Sect. 2. The application characteristics, the NoC platform and a motivational example are explained in Sect. 3. The proposed ERAM algorithm demonstrated in Sect. 4. Experimental results are presented in Sects. 5, and 6 concludes the paper.

2 Related Work and Novel Contribution

A few recent surveys on core mapping on NoCs for information on modern research and developments is presented below. Murali et al., proposed NMAP algorithm, which maps the cores onto NoC architectures with respect to bandwidth. It comprises three phases, mapping generation, based on the module that has the highest communication; determining the shortest path by using Dijkstra's algorithm and finally iterative enhancement by changing pairs of modules and recomputing the shortest paths [4]. In [5], Chou et al., explained mapping algorithm by using the ILP formulation. It reduces the inter tile network contention and resolves the scalability problem. The energy-aware and performance-aware incremental mapping technique proposed by Cho et al. [6], involves a real-time application to mapping on a homogenous multiple voltage-level NoC architecture. In [7], Jinho Lee et al. described task mapping and scheduling on a Multi-Processor SoC, and selected the communication path between processors. Liu et al. [8] proposed application mapping on a reconfiguration NoC architecture with respect to energy and reliability.

Spare core placement and its impact on system fault tolerance properties are discussed in Fault-Aware Resource Management (FARM) [9]. When core fails, the tasks of the associated failed core are migrated to spare cores. A spare core is selected based on functional metrics such as weighted manhattan distance, link contention count and system fragmentation factor. Zarandi [10] proposed Fault Aware Spare-core Allocation (FASA) algorithm, which improves performance. The placement of spare core is determined by considering the communication energy of temporary and permanent fault occurrences in the core.

Compared with previous related literature, this paper focus on energy- aware and reliability-aware mapping issues for NoC-based architectures and considers new applications with insignificant inter-processor communication overhead to be added to the system. This methodology was assessed by applying it to various benchmark applications. Finally the proposed algorithm can be applied to any reliable mapping application.

3 Preliminaries

3.1 Application Characteristics

The approaching applications are depicted by an application core graph (ACG), which is generated by [11]. Each ACG = (C, E) (see Fig. 1) is a coordinated diagram and contains the following [12].

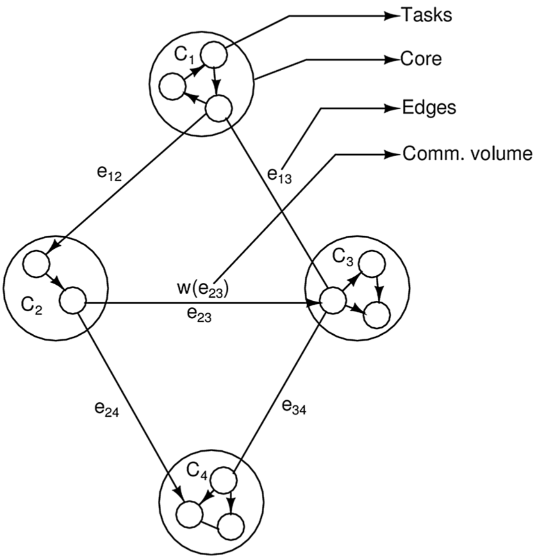

Fig. 1 Example of an application core graph

1. *Cores* Each Core $C_i \in$ C contains an arrangement of tasks from an offline task partition process. The tasks with the same core are assigned to the same PE.
2. *Edges* Each edge $e_{ij} \in$ corresponds to the communication between the two cores C_i and C_j. Core C_i is any neighbor of core C_j. Weights $W(e_{ij})$ characterize the communication volume (bits) between cores C_i and C_j.

3.2 NoC Platform

This platform is a two-dimensional (2D) mesh (mxn) NoC, which contains cores, routers, core interface and the link between cores. NoC cores are two types: (1) homogeneous and (2) heterogeneous. These cores are linked to the router via the network interface. Every router has five I/O ports, of which four ports are associated with the neighboring router and one is connected to the resident core. In the NoC, cores are three types: (1) regular core, (2) spare core and (3) manager core [13]. Regular cores run the task of a given application, they are two types: a. failed core and b. non-failed core. A spare core is an extra core that can be used when any processing core or manager core fails [14]. Manager cores manage and track all processing cores in the NoC, and perform the task migration if a processing core fails. These cores are similar in nature but have different functions at a given time.

The NoC prefers the mesh topology; this topology graph can be uniquely described using a directed graph TG(N, D). where 'N' represents a node or tile in the topology $\forall t_{xy} \in$ N, 't_{xy}' represents the xth row and yth column of the tile. 'D' represents the communication distance $\forall N_{ij} \in$ D and 'N_{ij}' denotes the distance between node (N_i) and node (N_j).

Communication between the routers is denoted as inter-tile communication and communication between the core and router is denoted as intra-tile communication. Rough mapping of the core graph to the NoC platform, and source-based, destination-based, and path-based communication is clearly shown in Fig. 2. Source-based communication is communication from the source to the destination. Destination-based communication is

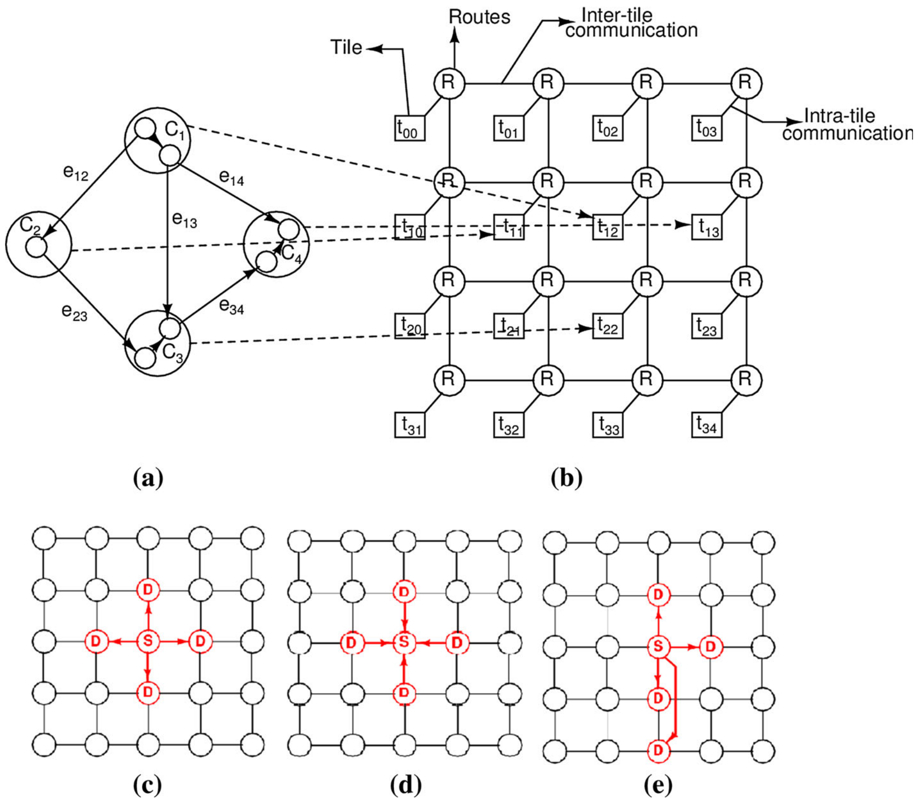

Fig. 2 **a** Application core graph, **b** 4 × 4 NoC platform, **c** source based communication, **d** destination based communication and **e** path based communication

communication from the destination towards the source. Path-based communication is shortest communication path from the source to the destination.

3.3 Motivational Example

To demonstrate the effect of the source-based, destination-based, and path-based network communication on packet latency, This work consider an example mapping configuration (see Fig. 3.) in a 5 × 5 mesh NoC in the following cases: with-out/with source-based communication (case 1 vs. case 2), with-out/with destination-based communication (case 3 vs. case 4), and with-out/with path-based communication (case 5 vs. case 6). Data transmission with 8 bits per packet and routing algorithm is chosen as XY. The injection rate and data transmission rate are the same for all configurations. Nearly 100 experiments were performed and the average packet latency was calculated. Latency is calculated as the time taken for a packet to be transferred from the source to the destination. The experimental results are as shown in Fig. 3, in which X-axis represents the packet injection rate and Y-axis represents the average packet latency. As seen in Fig. 3a–c represents source-based, destination-based and path-based communication, respectively. Furthermore, the frequency of occurrence of the path-based communication highly varied with those of the source-based and destination-based communication as the size increases. Results of a few

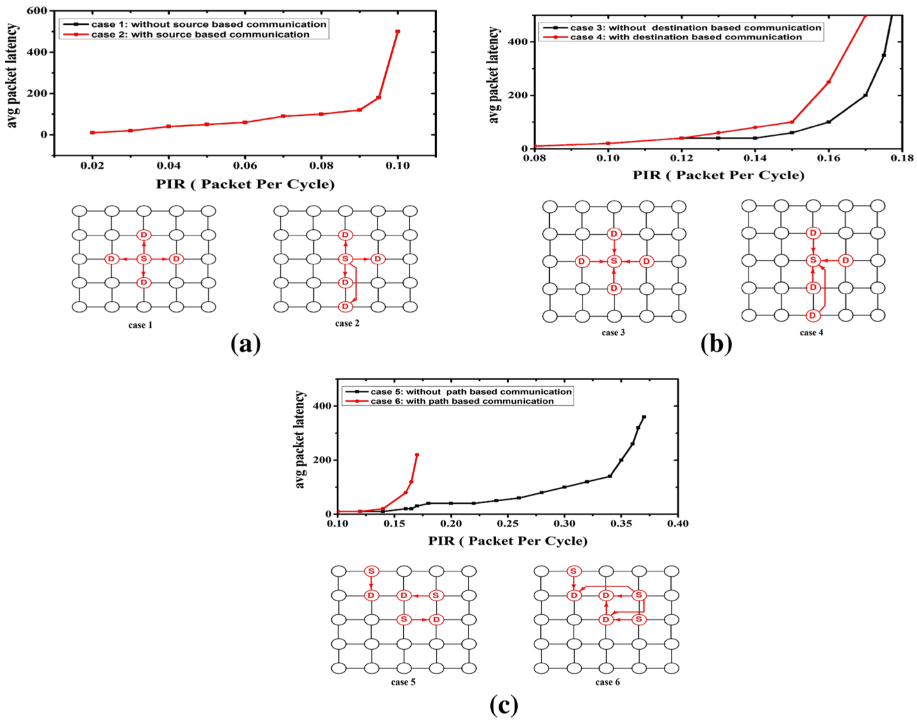

Fig. 3 The impact of **a** source-based, **b** destination-based and **c** path-based communication on average packet latency

experiments revealed that the ratio of path-based to source-based communication and the ratio of path-based to destination-based communication growth were linearly related with the size of the network traffic. Therefore, the focus on reducing the path-based communication because it has the most significant effect on packet latency. This can be achieved through the mapping process.

4 Energy and Reliability Aware Mapping

4.1 Problem Formulation

In this section, a core mapping algorithm is proposed. Proposed algorithm aims at minimizing the communication energy and path-based communication. More formally:

4.1.1 Weighted Communication Energy (WCE)

Energy is directly proportional to distance.

$$E \propto D \tag{1}$$

The distance function denotes the distance traveled from one position to the other if a square grid-path is followed. The distance between two positions is the sum of the differences of their equivalent modules. The distance between two cores (C_i, C_j), with C_i parameters (a_1, b_1) and C_j parameters (a_2, b_2) is given by,

$$Distance = |(a_1 - a_2)| + |(b_1 - b_2)| \tag{2}$$

Communication energy is directly proportional to the distance between nodes and is denoted by 'E_{ij}' [15].

$$E_{ij} \propto |(a_1 - a_2)| + |(b_1 - b_2)| \tag{3}$$

$$E_{ij} = e_{ij} \times \{|(a_1 - a_2)| + |(b_1 - b_2)|\} \tag{4}$$

where e_{ij} is a constant, which denotes the communication rate from C_i to C_j. E_{ij} is also called weighted communication energy [16].

Problem 1 Given an ACG (C, E) and NoC topology graph (N, D); Find a mapping function $\Omega : \mathrm{C} \rightarrow \mathrm{N}$ with $\forall C_i \in \mathrm{C}$ and calculate the total communication energy.

such that :

$$\begin{aligned} &\forall C_i \in C, \forall N_i \in N. \\ &\qquad \Omega(C_i) \in \mathrm{N}, \\ &\qquad C_i \neq C_j \Rightarrow \Omega(C_i) \neq \Omega(C_j) \\ &\qquad \forall e_{ij} \in \mathrm{e} \\ &\qquad \forall N_{ij} \in \mathrm{D} \end{aligned}$$

Let $\forall$ $C_i \in \mathrm{C}$ be mapped to some $t_{xy} \in \mathrm{N}$.

$$\text{then}\, t_{xy} = \Omega(C_i) \in N.$$

Communication energy is calculated using Eq. (4).

$$\begin{aligned} E_{ij} &= e_{ij} \times \{|(a_1 - a_2)| + |(b_1 - b_2)|\} \\ E_{ij} &= e_{ij} \times N_{ij} \end{aligned}$$

N_{ij} denotes the distance between node (N_i) and node (N_j). N_i parameters are (a_1, b_1) and N_j parameters are (a_2, b_2). e_{ij} denotes the communication rate from C_i to C_j.

$$\text{Total communication energy (E)} = \sum_{\forall C_i \in C} e_{ij} \times N_{ij}. \tag{5}$$

Problem 2 Given an Application Core Graph (C, E) and NoC Topology Graph (N, D); Find a mapping function Ω: $C \rightarrow \mathrm{N}$ with $\forall C_i \in C$ and the minimum communication energy. Which is given by

$$E = \sum_{\forall C_i \in C} e_{ij} \times N_{ij}$$

such that:

By using this definition, the problem of mapping can be formulated as follows. Given an ACG and an NoC topology that satisfy

$$\text{size}\,(\text{ACG}) \leq \text{size}\,(\text{NoC topology})$$
$$\text{map}(C_i) \in \text{N},$$

Problem 1 clearly explains mapping and communication energy. The minimum communication energy is obtained by,

$$f(E) = \left\{ \begin{array}{cc} \text{minE} & : \text{initial mapping} \\ \min\{e : e \in \text{E}\} & : \text{changing mapping} \end{array} \right\} \tag{6}$$

4.2 Proposed Mapping Heuristic

The proposed algorithm to map the given application cores on an NoC platform, which aims to minimize the communication energy and path-based communication.

Algorithm 1 CoreMapping

```
Input: Let C be the set of cores;
Let NoCP be the Given NoC Platform;
Output: Mapped NOC Platform ;

for $Iter \leq MaxIter$ do
    generate positions of empty cores randomly;
    map each c $\in$ C in orderly;
    calculate total communication energy;
    min $\leftarrow$ TotalCommunicationEnergy ;
    if $min > TotalCommunicationEnergy$ then
        min$\leftarrow$TotalCommunicationEnergy ;
        BestNoCMapping $\leftarrow$ NoCMapping;
    end
end
Return BestNocMapping;
```

Algorithm 2 FaultyCoreMapping

```
Input: Mapped NoC Platform;
Output: Best Faulty Core Mapped on NoC Platform i.e., BestFaultyMapping;

Let FC be the set of free cores available in a mapped NoC Platform;
BestFaultyMapping= {};
Identify the FaultyCore;
foreach $fc \in \{ FC \}$ do
    fc $\leftarrow$ FaultyCore ;
    FaultyMapping $\leftarrow$ NoC Platform where $FaultyCore$ mapped to fc ;
    TotalCommunicationEnergy $\leftarrow$ Compute total communication energy for
      FaultyMapping using Eq. (5);
    min $\leftarrow$ TotalCommunicationEnergy ;
    if $min > TotalCommunicationEnergy$ then
        min$\leftarrow$TotalCommunicationEnergy ;
        BestFaultyMapping $\leftarrow$ FaultyMapping;
    end
end
Return BestFaultyMapping;
```

Table 1 Benchmarks used in the simulation

Benchmark	Number of IP's	Application
DVOPD	32	Dual video object plane decoder
VOPD	16	Video object plane decoder
MPEG4	9	MPEG4 decoder
PIP	8	Picture in picture
MWD	12	Multi window display
mp3enc mp3dec	13	Mp3 encoder and Mp3 decoder
263enc mp3dec	12	H.263 encoder and Mp3 decoder
263dec mp3dec	14	H.263 decoder and Mp3 decoder
H.264	14	H.264decoder
HEVC	16	High efficiency video coding decoder
Freqmine	12	Data mining application
Swaption	15	compute portfolio prices using Monte-Carlo simulation
random1	16	Generated by TGFF
random2	16	Generated by TGFF

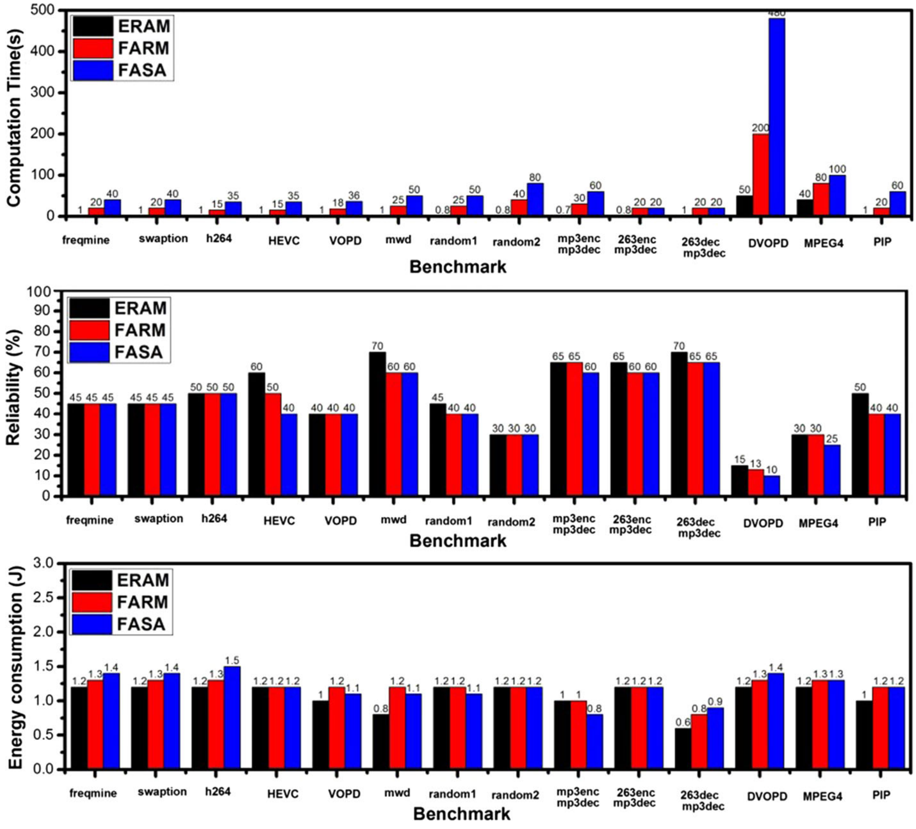

Fig. 4 Comparison of computation time, reliability and energy consumption among ERAM, FARM [9], and FASA [10]

By using Algorithm 1, cores can be efficiently mapped on an NoC platform, and the communication energy can be minimized. If faults occur at the processing core, faulty core tasks are migrated to the free core in the NoC platform, which should be located near the faulty core in the NoC platform. Faulty core mapping is explained in the Algorithm 2 [17, 18].

5 Experimental Results

The performance and communication energy of proposed mapping algorithm (ERAM—Energy and Reliability Aware Mapping) was assessed and compared with the existing mapping algorithms. The mapping pattern and computation time were calculated using a C++ program. Determined simulations were performed on a Noxim simulator [19, 20].

The contention impact on the core graph on a 5 × 5 NoC platform was evaluated. Several sets of synthetic applications were generated using TGFF [11]. In this experiment, number of cores were used 12–16 and edges are 10–50 [21]. Out of the fourteen commonly used application benchmarks, twelve were real applications and two were random

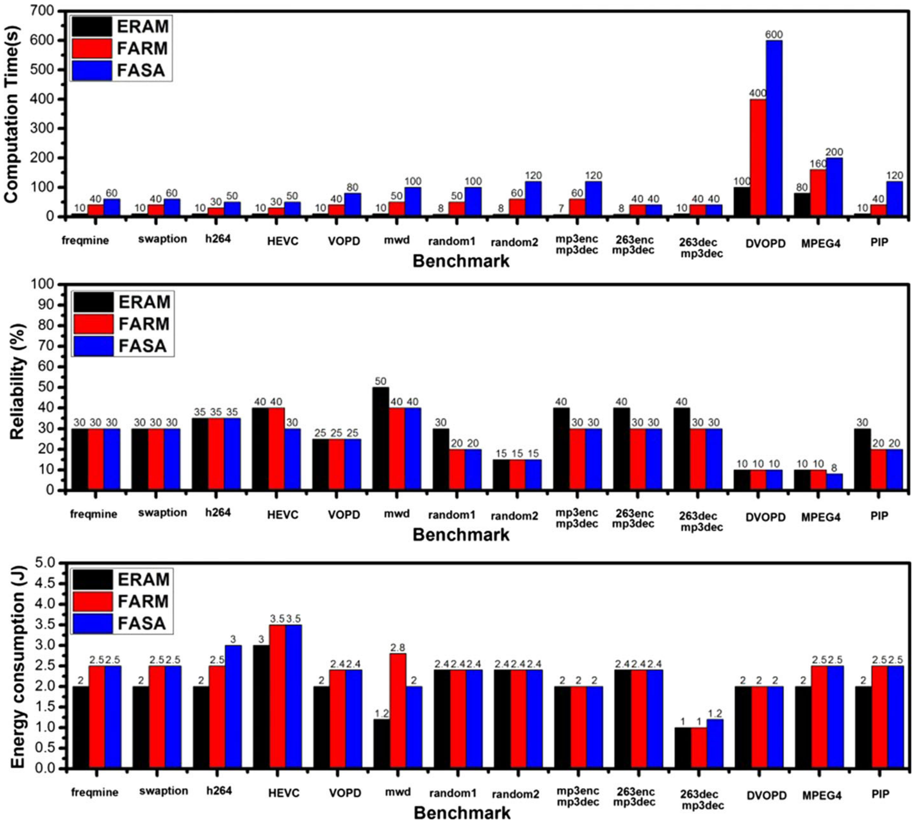

Fig. 5 Comparison of computation time, reliability and energy consumption among ERAM, FARM [9], and FASA [10] under a single fault condition

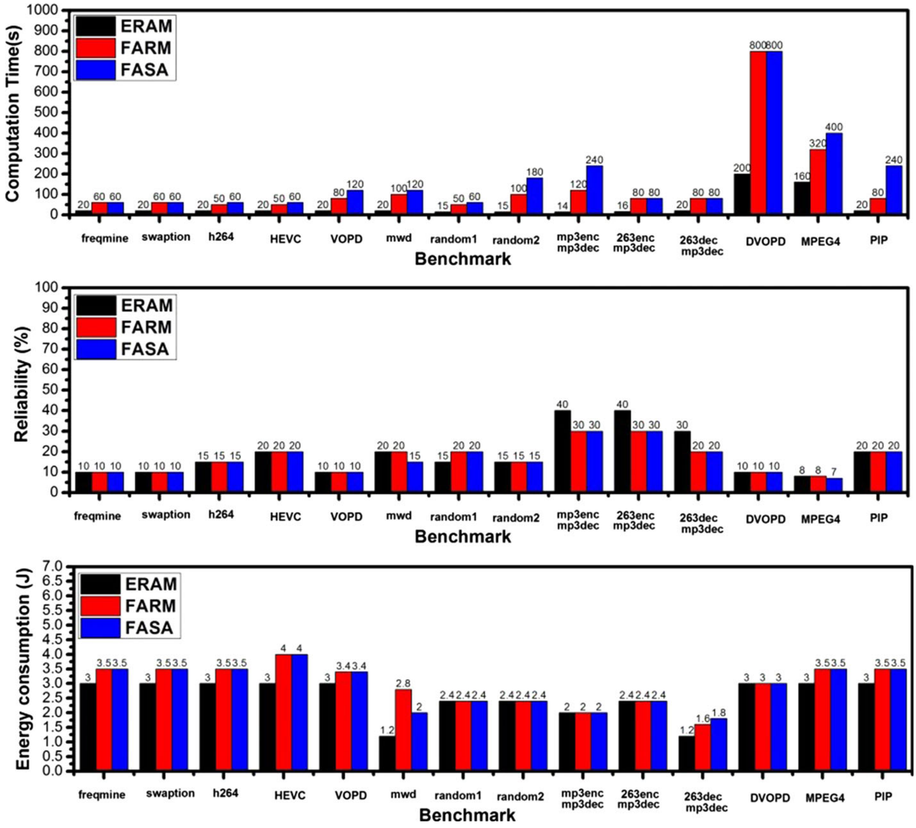

Fig. 6 Comparison of computation time, reliability and energy consumption among ERAM, FARM [9], and FASA [10] under a two faulty condition

Table 2 Communication energy and throughput comparison among ERAM, FARM [9], and FASA [10]

	ERAM is compared with FARM [9]					ERAM is compared with FASA [10]				
Number of edges	10	20	30	40	50	10	20	30	40	50
Communication energy conservation (%)	1	3	6	9	14	1	2	3	6	10
Throughput improvement (%)	18	20	22.8	24.2	26.8	8	13.7	16	18.2	20.8

benchmarks generated using TGFF as shown in Table 1 [22]. The computation times of ERAM and the existing algorithms (FARM [9] and FASA [10]) were compared to determine the optimal mapping as an indicator of the computational complexity. A comparison of reliability and energy consumption between ERAM and the existing algorithms are shown in Fig. 4. Reliability and energy consumption comparison results of ERAM and existing algorithms under a faulty core environment is as shown in Figs. 5, and 6.

As shown in Table 2, the communication energy conservation in ERAM has reduced by up to 14% compared to FARM [9] and 10% compared to FASA [10]. System throughput was improved on an average by 22.36% by using ERAM when compared to FARM [9] and 15.34% when compared to FASA [10]. These results clearly show that ERAM effectively reduces communication energy substantially improving the system throughput.

6 Conclusion

For the efficient design of future MPSoCs, an automatic mapping of cores onto NoCs is highly desirable. The proposed Energy and Reliability Aware Mapping (ERAM) algorithm was tested with both synthetic and real benchmarks. The results evidently implicate that there is a considerable reduction in communication energy and improvement in system reliability. Finally, efficient mapping results were obtained by using the proposed algorithm.

Acknowledgements The authors thanks the Department of Electronics and Information Technology, Ministry of Communication & IT, Government of India and Media Lab Asia, to fund this work under Visvesvaraya PhD scheme.

References

1. Benini, Luca, & Micheli, G. D. (2002). Networks on chips: A new SoC paradigm. *IEEE Computer*, *35*(1), 70–78.
2. Bertozzi, D., & Benini, L. (2004). Xpipes: A network-on-chip architecture for gigascale system on chip. *IEEE Circuits and Systems*, *4*(2), 18–31.
3. Beechu, N. K. R., et al. (2017). High-performance and energy-efficient fault-tolerance core mapping in NoC. *Sustainable Computing, Informatics and Systems*. https://doi.org/10.1016/j.suscom.2017.08.004.
4. Murali, S., & De Micheli, G. (2004) Bandwidth-constrained mapping of cores onto NoC architectures. In: *Proceedings design automation and test in Europe conference and exhibition*.
5. Chou, C.-L., & Marculescu, R. (2008) Contention-aware application mapping for network-on-chip communication architectures. *IEEE international conference on computer design* (pp. 164–169).
6. Chou, C.-L., Ogras, U. Y., & Marculescu, R. (2008). Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, *27*(10), 1866–1879.
7. Lee, J., Chung, M.-K., Cho, Y.-G., Ryu, S., Jung, H. A., & Kiyoung, C. (2013). Mapping and scheduling of tasks and communications on many-core SoC under local memory constraint. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, *32*(11), 1748–1761.
8. Liu, L., Wu, C., Deng, C., Yin, S., Wu, Q., Han, J., et al. (2014). A flexible energy- and reliability-aware application mapping for NoC-based recongurable architectures. *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, *23*, 2566–2580.
9. Chou, C.-L., & Marculescu, R. (2011). FARM: Fault-aware resource management in NoC based multiprocessor platforms. In: *Design automation & test in Europe conference & exhibition (DATE)*.
10. Khalili, Fatemeh, & Zarandi, Hamid R. (2013). A fault-tolerant core mapping technique in networks-on-chip. *IET Computers & Digital Techniques*, *7*(6), 238–245.
11. Task graphs for free (TGFF). http://ziyang.eecs.umich.edu/∼ dickrp/tgff/
12. Naresh Kumar Reddy, B., Vasantha, M. H., Nithin Kumar, Y. B., & Sharma, D. (2015). A fine grained position for modular core on NoC. In: *IEEE international conference on computer, communication and control (IC4)* (pp. 1–4).
13. Nollet, V., Marescaux, T., & Verkest, D. (2004). Operating system controlled network on chip. In: *Design automation conference* (pp. 256–259).
14. Becchu, N. K. R., et al. (2017). System level fault-tolerance core mapping and FPGA-based verification of NoC. *Microelectronics Journal*, *70*, 16–26.
15. http://grokbase.com/t/lucene/mahout-dev/082festv1e/weighted-manhattan-distance-metric

16. Naresh Kumar Reddy, B., Vasantha, M. H., Nithin Kumar, Y. B., & Sharma, D. (2015). Communication energy constrained spare core on NoC. In: *6th international conference on computing, communication and networking technologies (ICCCNT)* (pp. 1–4).
17. Yuankun, X., & Bogdan, P. (2016). Improving NoC performance under spatio-temporal variability by runtime reconfiguration: a general mathematical framework. In: *2016 tenth IEEE/ACM international symposium on networks-on-chip (NOCS)*, IEEE.
18. Bogdan, P., & Xue, Y. (2015) Mathematical models and control algorithms for dynamic optimization of multicore platforms: a complex dynamics approach. In: *Proceedings of the IEEE/ACM international conference on computer-aided design (ICCAD '15).*
19. Noxim the NoC simulator. https://github.com/davidepatti/noxim
20. Naresh Kumar Reddy, B., Vasantha, M. H., & Nithin Kumar, Y. B. (2016). A gracefully degrading and energy-efficient fault tolerant NoC using spare core. In: *2016 IEEE computer society annual symposium on VLSI (ISVLSI 2016)* (pp. 146–151), Pennsylvania, USA.
21. Railing, B. P., Hein, E. R., & Conte, T. M. (2015). Contech: Efficiently generating dynamic task graphs for arbitrary parallel programs. *ACM Transactions on Architecture and Code Optimization (TACO)*, *12*(2), 25.
22. Xue, Y., & Bogdan, P. (2016). Scalable and realistic benchmark synthesis for efficient NoC performance evaluation: A complex network analysis approach. In: *2016 international conference on hardware/software codesign and system synthesis (CODES+ ISSS)*, IEEE.

Naresh Kumar Reddy Beechu received the bachelor of engineering degree in electronics and communication in 2010 from the S.V. University, India, and master's of engineering degree in embedded system in 2012 from the K.L. University, India. He is currently pursuing Ph.D. in NIT Goa, India. His research interests focus on fault-tolerant and networks-on-chip and embedded systems, on-chip multiprocessors, reconfigurable systems. He is coauthor of several scientific papers on networks-on-chip, design methodologies for systems-on-chip, embedded system.

Vasantha Moodabettu Harishchandra received the masters of engineering degree in electronics and communication in 2003 from the IITM, India, and Ph.D in 2014 from the NIT Surathkal, India. He is currently Assistant professor and HOD of Electronics department in NIT Goa, India. His research interests focus on Low voltage, Low power analog mixed signal circuits, Continuous-time filter Circuits, System on Chip, fault-tolerant and networks-on-chip and embedded systems.

Nithin Kumar Yernad Balachandra received the masters of engineering degree in electronics and communication in 2003 from the NIT Surathkal, India, and Ph.D. in 2014 from the IIT Kharagpur, India. He is currently Assistant professor in Electronics departmentin NIT Goa, India. His research interests focus on Analog and Mixed Signal Design, System on Chip, fault-tolerant and networks-on-chip and embedded systems.